

# E8 Emulator

Additional Document for User's Manual

R0E000080KCE00EP27

Renesas Microcomputer Development Environment System  
M16C Family / R8C/Tiny Series  
Notes on Connecting the R8C/2A, R8C/2B, R8C/2C and R8C/2D

User's Manual

Rev.1.00  
February 16, 2007

RenesasTechnology  
[www.renesas.com](http://www.renesas.com)

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.  
The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.  
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/ or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

## Contents

Section 1	Specifications of the E8 Emulator .....	1
Section 2	Connecting the Emulator with the User System .....	3
Section 3	Pin Assignments of the E8 Connector .....	5
Section 4	Example of E8 Connection .....	7
Section 5	Notes on Using the E8 Emulator.....	11
Section 6	Setup the Debugger .....	18
Section 7	Applicable Tool Chain and Partner Tools .....	21

This user's manual is applicable to the E8 emulator software V.2.09 Release 02 or later.



## Section 1 Specifications of the E8 Emulator

Table 1.1 shows the specifications of the R8C/2A, R8C/2B, R8C/2C and R8C/2D Groups E8 Emulator.

Table 1.1 Specifications of the R8C/2A, R8C/2B, R8C/2C and R8C/2D Groups E8 Emulator

Target MCU	M16C Family R8C/Tiny Series R8C/2A, R8C/2B, R8C/2C and R8C/2D Groups
Usable operating mode	Single-chip mode
Break function	- Address match break, 4 points, or Address match break, 2 points + Data condition break, 1 point - PC break (up to 255 points) - Forcible break
Trace function	Latest 4 branch instructions
Flash memory programming function	Available
User interface	1-line clock asynchronous serial interface (communicating via MODE pin) * UART1 function cannot be used in a user program
MCU resource to be used	- ROM size: 2 KB * - Stack 8 bytes - Address match interrupt - UART 1 function
Emulator power supply	Unnecessary (USB bus powered, power supplied from the PC)
Interface with host machine	USB (USB 1.1, full speed) * Also connectable to host computers that support USB 2.0
Power supply function	Can supply 3.3V or 5.0V to the target board (300mA, max)
Power voltage	3.0--5.5V (f(XIN)=20MHz) 2.7--5.5V (f(XIN)=10MHz)

\* Varies depending on the device. For detail, see "Section 5 Note on Using the E8 Emulator".

Table 1.2 shows the operating environment of the E8 Emulator.

Table 1.2 Operating Environment

Temperature	Operating	: 10°C to 35°C
	Storage	: -10°C to 50°C
Humidity	Operating	: 35% RH to 80% RH, no condensation
	Storage	: 35% RH to 80% RH, no condensation
Vibration	Operating	: 2.45 m/s <sup>2</sup> max.
	Storage	: 4.9 m/s <sup>2</sup> max.
	Transportation	: 14.7 m/s <sup>2</sup> max.
Ambient gas	No corrosive gas	



## Section 2 Connecting the Emulator with the User System

Before connecting an E8 emulator with the user system, a connector must be installed in the user system so that a user system interface cable can be connected. When designing the user system, refer to Figure 3.1, Pin Assignments of the E8 Connector, and Figure 4.1, Example of E8 connection, shown in this manual.

Before designing the user system, be sure to read the E8 emulator user's manual and the hardware manual for related MCUs.

Table 2.1 shows the recommended connector for the emulator.

Table 2.1 Recommended Connector

	Type Number	Manufacturer	Specifications
14-pin connector	7614-6002OO*	3M Limited	14-pin straight type

\*OO indicates coat specification

Connect pins 2, 4, 6, 10, 12, and 14 of the user system connector to GND firmly on the PCB. These pins are used as electrical GND and to monitor the connection of the user system connector. Note the pin assignments of the user system connector.

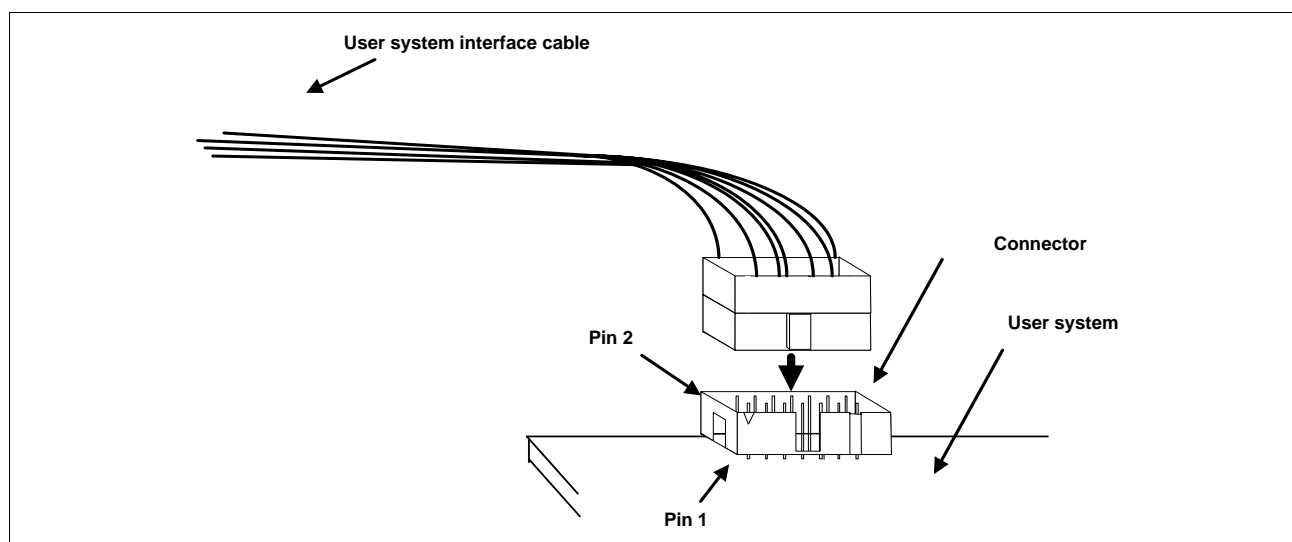


Figure 2.1 Connecting the User System Interface Cable to the User System

- Notes:
1. Do not place any components within 3 mm of the connector.
  2. When using the E8 emulator as a programmer, connect the E8 emulator to the user system in the same way.



## Section 3 Pin Assignments of the E8 Connector

Figure 3.1 shows the pin assignments of the connector.

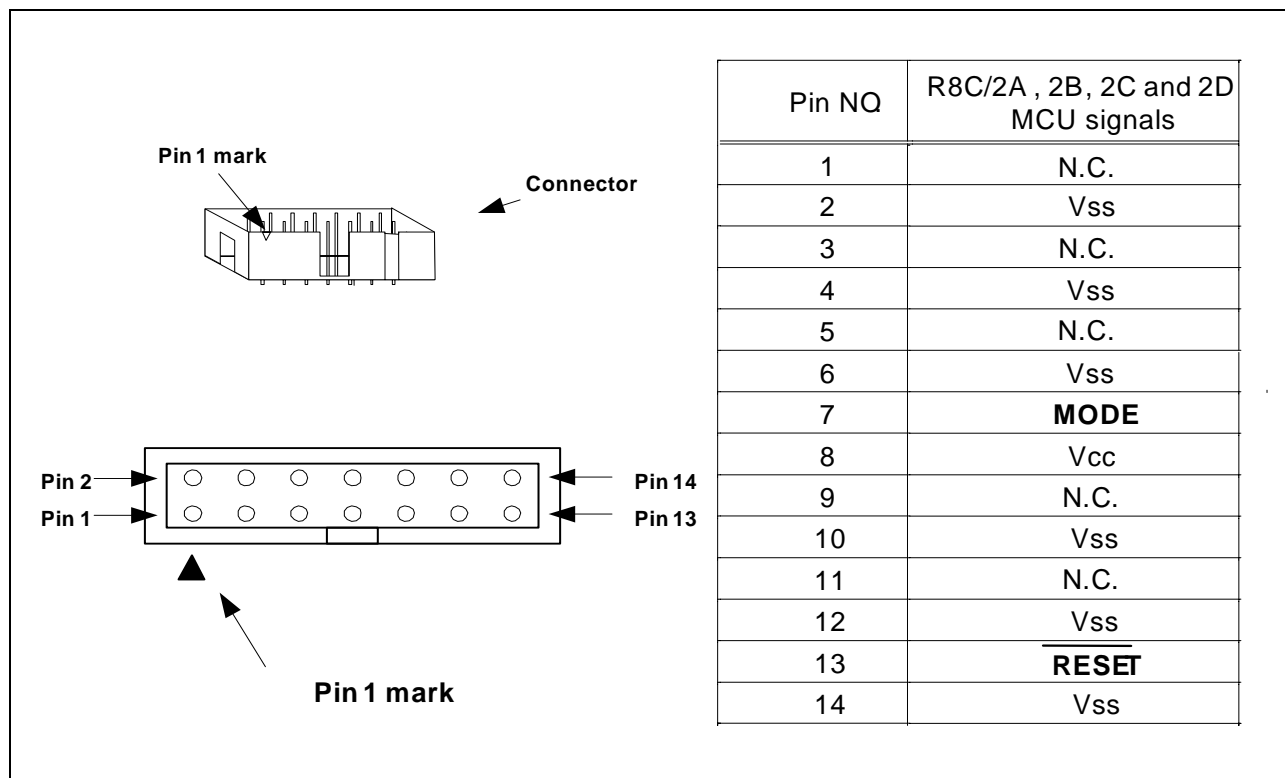


Figure 3.1 Pin Assignments of the E8 Connector

Note:

Pin 14 is used to check for the connection between the E8 and the user system, and pin 4 is connected to the internal circuit. These pins are not directly connected to Vss inside the E8. Besides pins 4 and 14, make sure that other pins 2, 6, 10, and 12 are connected to Vss.



## Section 4 Example of E8 Connection

Figure 4.1 shows the connecting example.

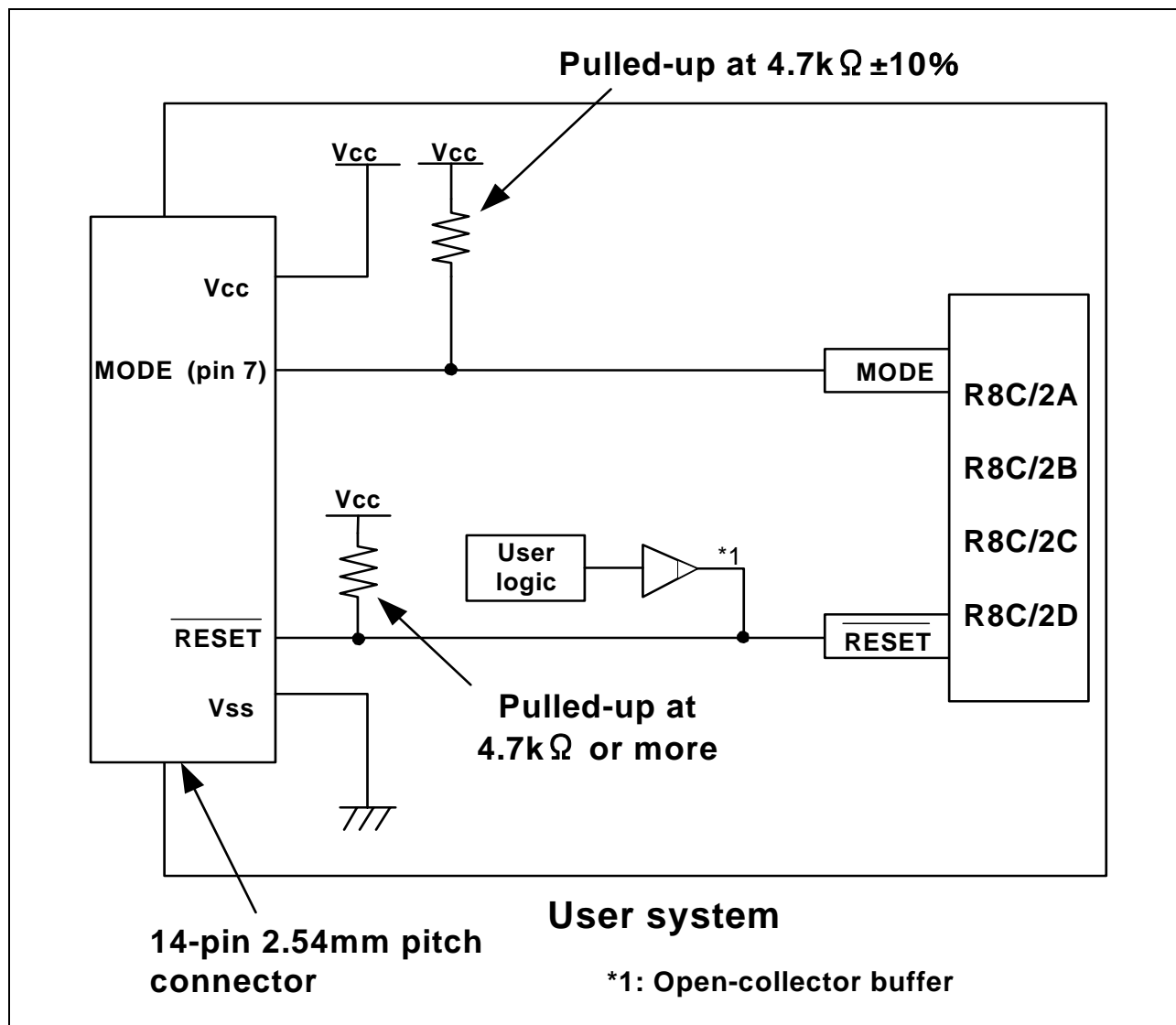


Figure 4.1 Example of E8 Connection

When using the emulator as a programmer, the specification of connection between the E8 and the MCUs is the same as shown in Figure 4.1.

Notes: 1. The E8 emulator uses the MODE pin for the MCU control and the forcible break control. Pull up the E8 emulator and MCU pins to connect the E8 emulator.

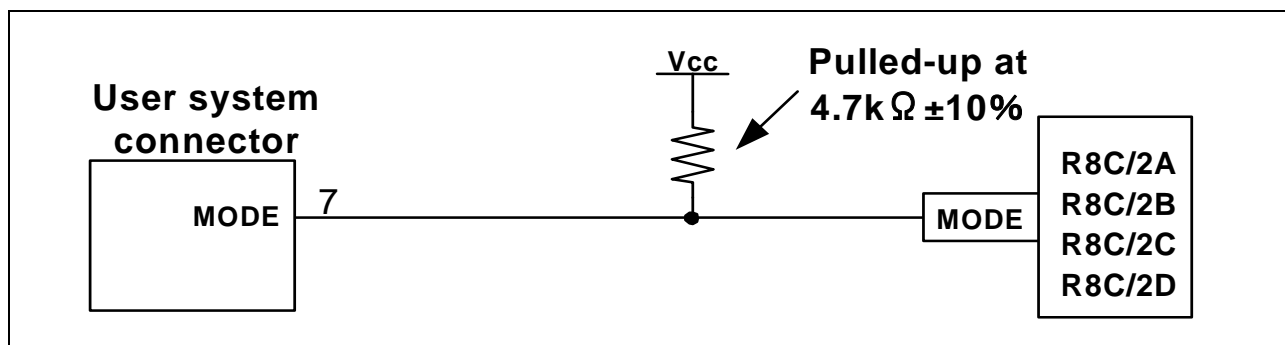


Figure 4.2 Connection of E8 Emulator and MODE Pin

2. The  $\overline{\text{RESET}}$  pin is used by the E8 emulator. Therefore use an open-collector output buffer or a CR reset circuit as the reset circuit of the user system. The recommended pull-up value is 4.7kΩ or more. The MCU can be reset by outputting “L” from the E8 emulator. However, if a reset circuit on the user system is the H-output type reset IC, it cannot be set “L” in the reset circuit on the user system and the E8 emulator will not operate normally.

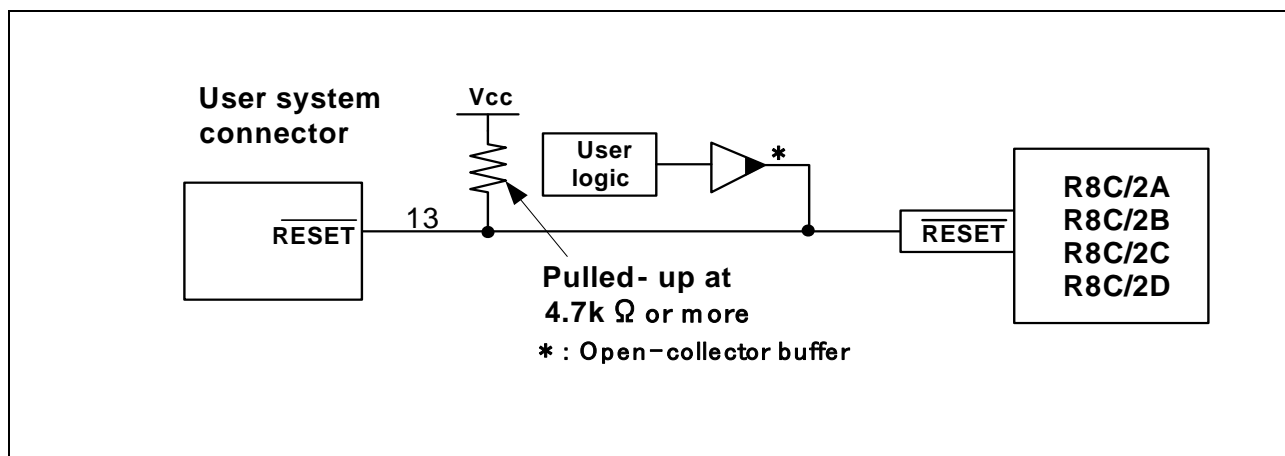


Figure 4.3 Example of a Reset Circuit

3. Connect Vss and Vcc with the Vss and Vcc of the MCU, respectively.
4. Connect nothing with N.C.
5. The amount of voltage permitted to input to Vcc must be within the guaranteed range of the microcomputer.
6. Pin 14 is used to check for the connection between the E8 and the user system, and pin 4 is connected to the internal circuit. These pins are not directly connected to Vss inside the E8. Besides pins 4 and 14, make sure that other pins 2, 6, 10, and 12 are connected to Vss.

7. Figure 4.4 shows the interface circuit in the E8 emulator. Use this figure as a reference when determining the pull-up resistance value.

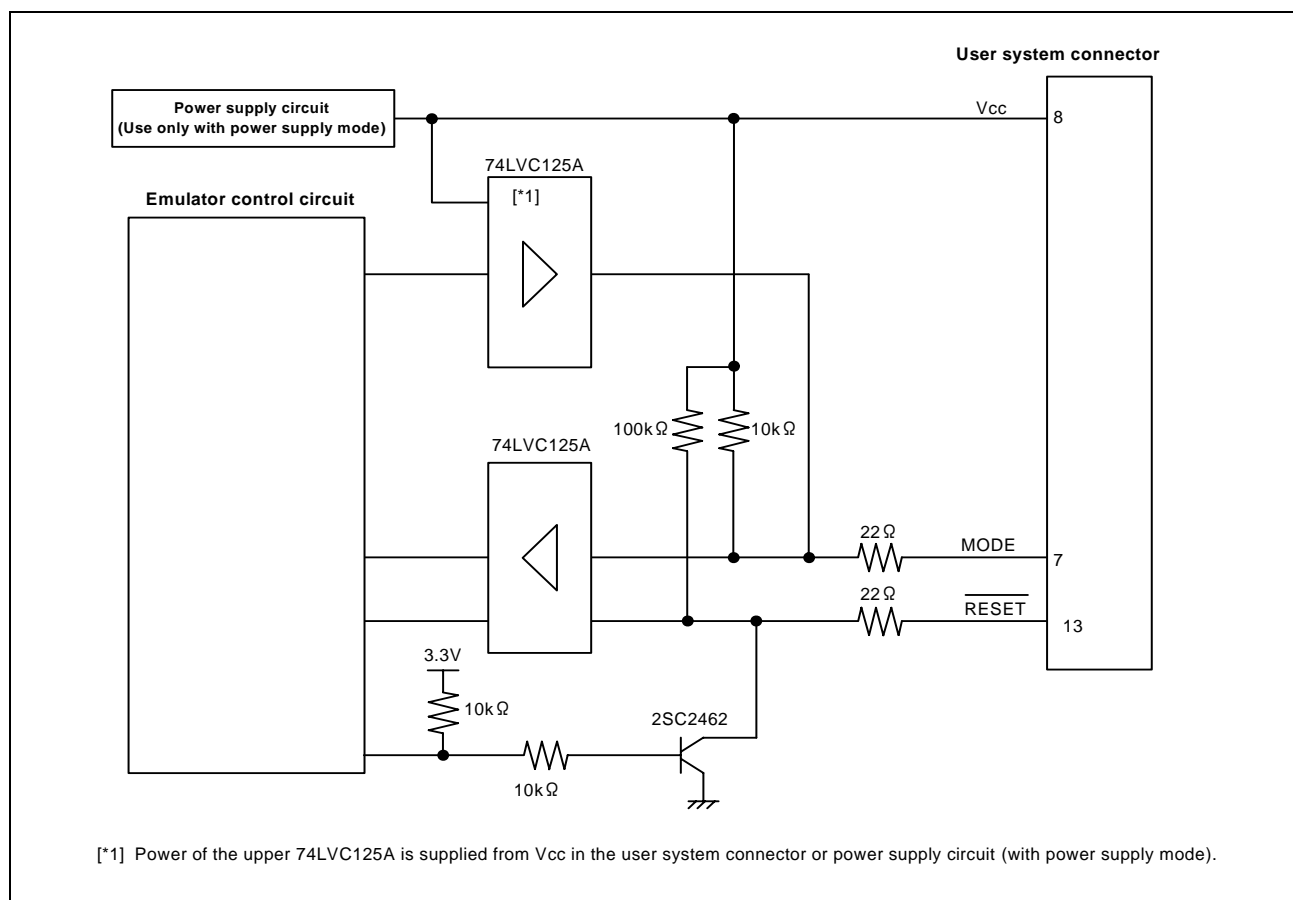


Figure 4.4 Interface Circuit in the E8 Emulator (Reference)



## Section 5 Notes on Using the E8 Emulator

### 1. Program area for the E8 emulator

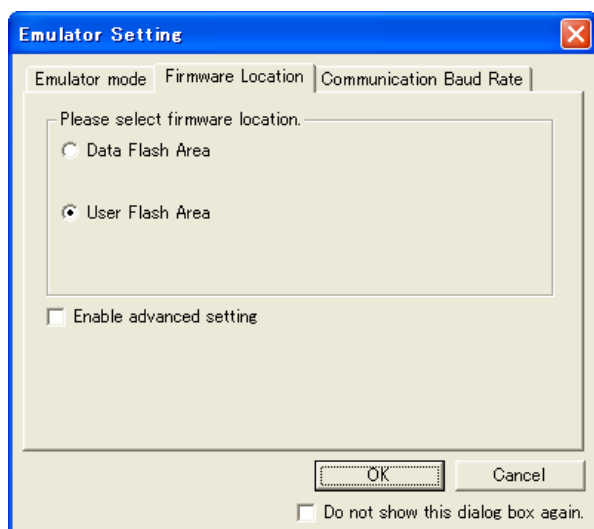
Table 5.1 lists the program area for the E8 emulator.

Do not change this area. If this area is changed, the E8 emulator will not operate normally. In this case, disconnect to the debugger and then reconnect.

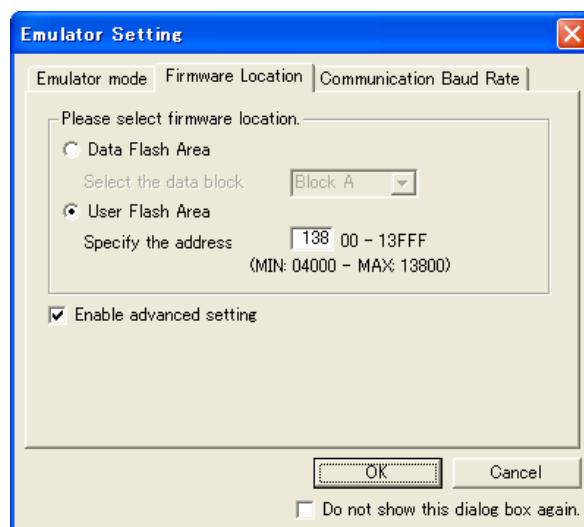
Table 5.1 Program Area for the E8 Emulator

Group	Type Number	ROM Size		Program Area for E8 Emulator	
		Programming Area	Data Area	Vector Area	ROM Area (Default Area)
R8C/2A	R5F212A7	48KB	-	FFE4h-FFE7h, FFE8h-FFEBh, FFECCh-FFEFh, FFF4h-FFF7h, FFF8h-FFFBh FFFCh-FFFEh	-
	R5F212A8	64KB	-		2KB of the ROM area [*1] (User Flash Area 13800h-13FFFh)
R8C/2B	R5F212B7	48KB	2KB		-
	R5F212B8	64KB	2KB		2KB of the ROM area [*1] (Data Flash Area 2400h-2BFFh) or (User Flash Area 13800h-13FFFh)
R8C/2C	R5F212C7	48KB	-		-
	R5F212C8	64KB	-		2KB of the ROM area [*1] (User Flash Area 13800h-13FFFh)
R8C/2D	R5F212D7	48KB	2KB		-
	R5F212D8	64KB	2KB		2KB of the ROM area [*1] (Data Flash Area 2400h-2BFFh) or (User Flash Area 13800h-13FFFh)

- \*1: When starting up the debugger, the dialog box shown in Figure 5.1 is displayed. Specify an area which is not used in the user system. If [Enable advanced setting] check box is checked, the area used by the emulator can be specified with the address. (Figure 5.2).



**Figure 5.1 [Firmware Location] tab (1) in [Emulator Setting] Dialog Box**



**Figure 5.2 [Firmware Location] tab (2) in [Emulator Setting] Dialog Box**

2. When the E8 emulator system is initiated, it initializes the general registers and part of the flag registers as shown in Table 5.2.

**Table 5.2 Register Initial Values at Emulator Power-On**

Status	Register	Initial Value
Emulator Power-On	PC	Reset vector value in the vector address table
	R0 to R3 (bank 0, 1)	0000h
	A0, A1 (bank 0, 1)	0000h
	FB (bank 0, 1)	0000h
	INTB	0000h
	USP	0000h
	ISP	05FFh
	SB	0000h
	FLG	0000h

3. The E8 emulator communicates with the MCUs by using the MODE and  $\overline{\text{RESET}}$  pins.
4. The E8 emulator uses up to four-word stack pointer when a user program breaks. Accordingly, reserve the four-word addresses for the stack area.
5. Peripherals used by the E8 emulator program  
 UART1 is used by the E8 emulator. Do not use UART1 by the user program.

## 6. SFR used by the program for the E8 emulator

The SFR listed in Table 5.3 is used by the program for the E8 emulator as well as the user program. Do not change the value in the memory window, etc., by other than the user program. Note that although the SFR can be changed during the user program execution, the changed value cannot be read at break.

The SFR listed in Table 5.4 is used by the program for the E8 emulator, not user program. Do not change values of these SFRs, otherwise the E8 cannot be controlled. Note that UART1 transmit interrupt control register S1TIC and UART1 receive interrupt control register S1RIC always read out the value of using the emulator.

The SFRs listed in Tables 5.3 and 5.4 are not initialized by selecting [Debug] -> [Reset CPU] or with the RESET command. If their contents are referred to, a value that has been set in the program for the E8 emulator will be read.

Table 5.3 SFR Used by Program for E8 Emulator (1)

Address	Register	Symbol	Bit
000Ah	Protect register	PRCR	Bit 0 [*1]
0023h	High-speed on-chip oscillator control register 0	FRA0	Bit 0
0024h	High-speed on-chip oscillator control register 1	FRA1	All bits
01B3h	Flash memory control register 4	FMR4	Bit 7

\*1 From the E8 emulator software V.2.08 Release 00, Bit 1 of the protect register can change the value in other programs as well as the user program.

Table 5.5 SFR Used by Program for E8 Emulator (2)

Address	Register	Symbol	Bit	Notes on using the E8 emulator
0010h - 0012h	Address match interrupt register 0	RMAD0	All bits	[*2]
0013h	Address match interrupt enable register	AIER	All bits	[*2]
0014h - 0016h	Address match interrupt register 1	RMAD1	All bits	[*2]
00A8h	UART1 transmit/receive mode register	U1MR	All bits	[*2]
00A9h	UART1 bit rate register	U1BRG	All bits	[*2]
00AAh, 00ABh	UART1 transmit buffer register	U1TB	All bits	[*2]
00ACh	UART1 transmit/receive control register 0	U1C0	All bits	[*2]
00ADh	UART1 transmit/receive control register 1	U1C1	All bits	[*2]
00AEh, 00AFh	UART1 receive buffer register	U1RB	All bits	[*2]
00F8h	Port mode register	PMR	Bit 4	[*3]

\*2 Do not change the value of the register.

\*3 Do not change the value of the bits listed above. When operating this register, change it by a bit operating instruction, etc. in order to avoid changing the value of relevant bits.

## 7. Interrupts used by the E8 emulator program

The BRK instruction interrupt, address match interrupt, single-step interrupt, and address break interrupt are used by the E8 emulator program. Therefore, make sure the user program does not use these interrupts.

The E8 emulator changes these interrupt vector values to the values to be used by the emulator. It is not a problem if the interrupt vector values are written in the user program.

## 8. Debugging of a watchdog timer

During the program for the E8 emulator operation, the program for the E8 emulator refreshes the watchdog timer.

If a memory is accessed by the memory reference or modification, the watchdog timer will be refreshed by the program for the E8 emulator. Note that the operation timing is different from the actual one.

## 9. ID code of flash memory

This is the function of the MCUs which prevents the flash memory from reading out by other than the user. The 7 bytes ID code in Table 5.5 written to the flash memory of the MCUs have to match with the ID code displayed in Figure 5.3 [ID Code verification] dialog box at the debugger startup, otherwise the debugger cannot be started up. Note that when the ID code is FFh, FFh, FFh, FFh, FFh, FFh, FFh, ID code is considered not set up. In this case, the ID code is automatically authenticated and [ID Code verification] dialog box is not displayed.

When debugging in [Erase Flash and Connect] mode or [Keep Flash and Connect] mode, the ID code, FFh, FFh, FFh, FFh, FFh, FFh, FFh is written into the ID code area regardless of the contents of the user program. In [Program Flash] mode, the value to be written in the ID code area depends on the contents of the user program.

Table 5.5 ID Code Storage Area of R8C/2A, 2B, 2C and 2D

Address	Description
FFDFh	First byte of ID code
FFE3h	Second byte of ID code
FFEBh	Third byte of ID code
FFEFh	Fourth byte of ID code
FFF3h	Fifth byte of ID code
FFF7h	Sixth byte of ID code
FFFBh	Seventh byte of ID code

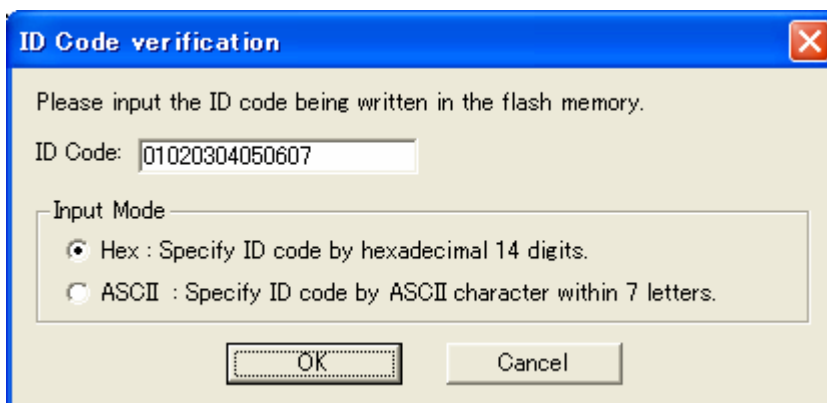


Figure 5.3 [ID Code verification] Dialog Box

### [Note on Program Flash mode]

When the ID code is specified by the -ID option of the lmc30, download the MOT file or HEX file. When the X30 file is downloaded, the ID code is not effective. When downloading the X30 file, specify the ID code using an assembler directive command such as “.BYTE”. The file to which the ID code specified by the assembler directive command “.ID” is output varies depending on the version of the assembler. For details, refer to the user’s manual of the assembler.

10. Operation clock during a break

During a user program break, the emulator operates changing the CPU clock to the internal high-speed on-chip oscillator (approx. 8 MHz). However, the peripheral features operate with the clock specified by the user program.

11. Reset

The reset vector is used by the E8 emulator program. If the MCU is reset while executing the user program, control is transferred to the E8 emulator program and the user program is made to stop. Do not use the voltage monitor 0 reset, the voltage monitor 1 reset, voltage monitor 2 reset, watchdog timer reset and software reset, otherwise the E8 emulator will not operate normally.

12. Memory access during emulation execution

When referring or modifying the memory contents, the user program is temporarily halted. For this reason, the real-time emulation cannot be performed. When the real-time emulation is necessary during the program operation, firstly disable the automatic update in the watch window or fix the display in the memory window so that the memory access will not occur during execution.

13. When the E8 does not supply power to the user system, the E8 emulator consumes the power voltage of the user system from several mA to over 10 mA. This is because the user power supply drives one 74LVC125A to make the communication signal level match the user system power supply voltage.

14. When debugging, the flash memory is frequently re-written by the E8 emulator. Therefore, do not use an MCU that has been used for debugging.

Also, as the program for the E8 emulator is written into the MCU while debugging, save the contents of the MCU's flash memory that have been used for debugging. Do not use them as the ROM data for products.

15. Reserved area

The addresses not specified in the Hardware Manual for R8C/2A, R8C/2B, R8C/2C and R8C/2D Groups are reserved area. Do not change the contents. Otherwise, the E8 emulator cannot be controlled.

16. Debugging in the stop mode or wait mode

When using the stop mode or wait mode on a user program, do not operate the window until the program stops at the breakpoint by setting the breakpoint at the processing unit where the stop mode or wait mode is cancelled. In addition, firstly disable the automatic update in the watch window or fix the display in the memory window so that the memory access will not occur during execution.

When the program is stopped forcibly, and the memory is referred or altered in the stop or wait mode, the stop or wait mode will be cancelled.

17. Peripheral I/Os during a break

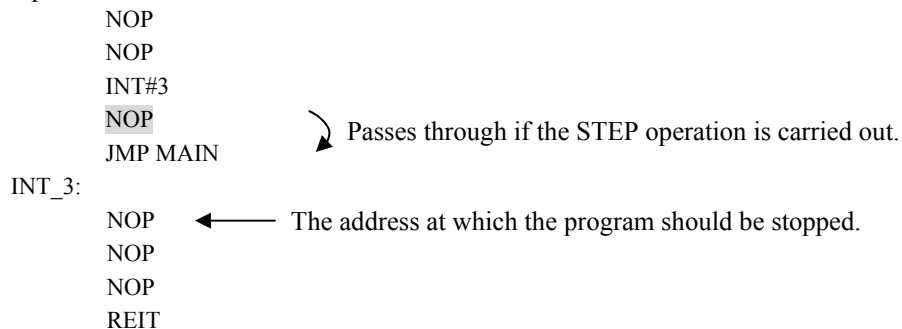
During a break, although interrupts are not accepted, peripheral I/Os continue to be operated. For example, a timer interrupt is not accepted although counting a timer is continued when a user program is stopped by a break after operating a timer.

## 18. Exceptional step operation

### a) Software-interrupt instruction

STEP operation cannot be performed by continuously executing the internal processing of instructions (undefined, overflow, BRK, and INT) which generates a software interrupt.

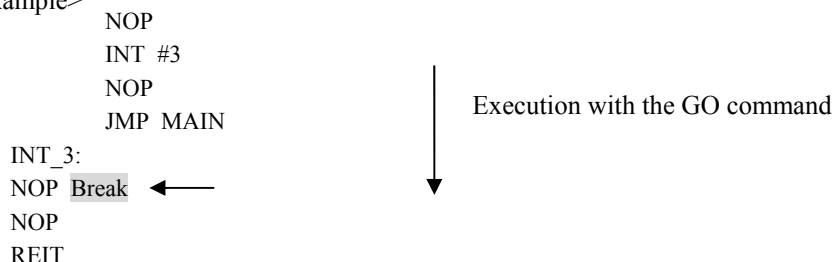
<Example> INT instruction



### b) INT instruction

Debugging of the program using the INT instruction should be used with the GO command by setting a PC break for the internal processing of the INT instruction.

<Example>



## 19. "Go to cursor" function

The "Run to cursor" function is realized by using an address match break. Therefore, when you execute the "Go to cursor" command, all the address match breaks and the hardware breaks you set become invalid, while all the PC breaks remain valid.

## 20. Note on PC break point

When downloading a user program after changing it, the set address of PC break may not be corrected normally depending on the changes. After downloading a user program, please check the setting of PC break by event point window and reset it.

## 21. Note on setting the break condition of the hardware breaks

When selecting the normal address bus (selecting [Address] radio button of [Break condition] dialog box) as a condition for the address bus, do not set the following addresses. Otherwise, a malformed break may occur.

- Address in the interrupt vector table
- Address set in the interrupt vector table (interrupt routine start address)
- Branched address of the branching instruction

Both fixed vector table and variable vector table are included with the interrupt vector table above.

22. Note on debugging in CPU rewrite mode

CPU rewrite mode can be executed only for the data area. If the CPU rewrite mode is executed for the program area, E8 emulator will run out of control.

When rewriting the data area, do not halt the user program after setting the CPU rewrite mode until releasing it. If you do so, the E8 emulator may run out of control. Cancel the automatic renewal in the watch window in advance and select fixing display in the memory window to prevent a memory access from occurring while executing the user program.

To check the data after executing the CPU rewrite mode, halt the program after releasing the CPU rewrite mode and see the memory window etc.

23. Notes on rewriting the flash memory

Do not reset the MCUs when rewriting the flash memory. The flash memory is completed to rewrite when the “Flash memory write end” is displayed in the output window of the High-performance Embedded Workshop. If the MCU is reset during rewriting the flash memory, the user program or the program for the E8 emulator may be disrupted

The followings indicate when the flash memory rewrite occurs.

- When downloading the user program
- After the user program starts with setting up PC break on the flash memory
- After the user program starts with canceling PC break on the flash memory
- After the user program starts with rewriting the value of flash memory in the memory window

24. Notes on the E8 emulator power supply

When writing reliability required program with the E8 emulator under the mass production, do not use the E8 emulator power supply function. Separately supply appropriate voltage according to the MCU programming to the user system. Since the supplied voltage from the E8 emulator depends on the quality of the USB power supply of the PC, its precision is not guaranteed.

25. Notes on the emulator setup switch

Use the emulator setup switch with the factory setting (upper side “1”).

## Section 6 Setup the Debugger

### 1. Emulator Setting dialog box

[Emulator Setting] dialog box is provided for setting the items that need to be set when the debugger starts up. The contents set from this dialog box other than “Power supply” are also effective the next time the debugger starts.

When starting up the debugger first time after creating the new project work space, [Emulator Setting] dialog box is displayed with the Wizard.

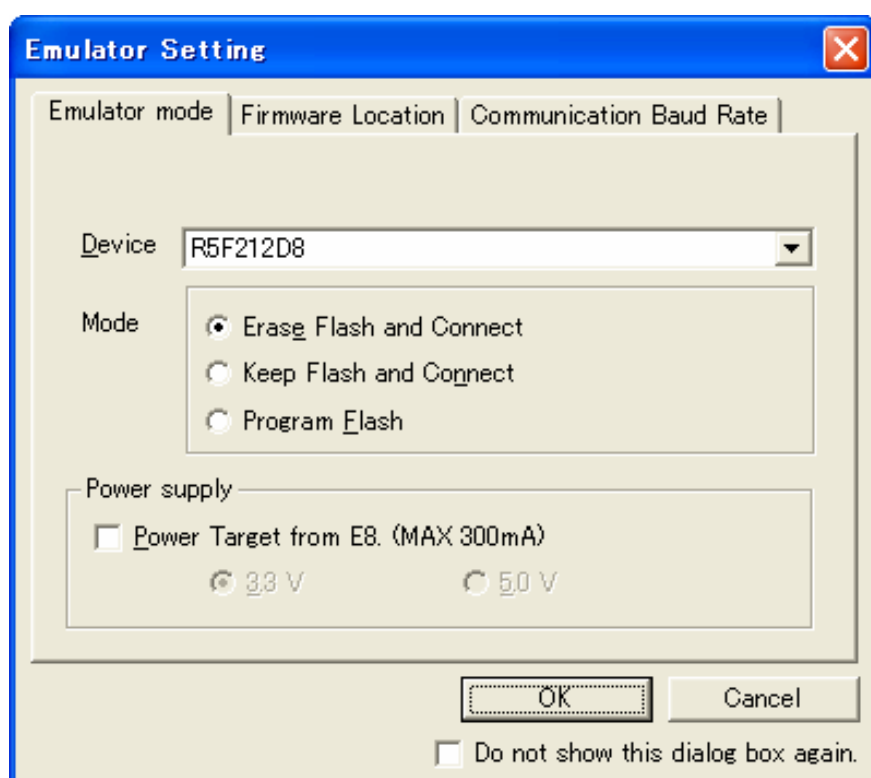


Figure 6.1 [Emulator Setting] Dialog Box.

To keep the [Emulator Setting] dialog box closed the debugger is started next time, check "Do not show this dialog box again." at the bottom of the [Emulator Setting] dialog box. You can open the [Emulator Setting] dialog box using either one of the following methods:

- After the debugger gets started, select Menu - [Setup] -> [Emulator] -> [Emulator Setting (E)...].
- Start Debugger while holding down the Ctrl key.

When you check the "Do not show this dialog box again.", the E8 does not supply power to the user system.

## 2. Emulator mode Tab

The selection of the device, the specification of the mode, and the setting of the power supply are done in the [Emulator mode] tab of the [Emulator Setting] dialog box.

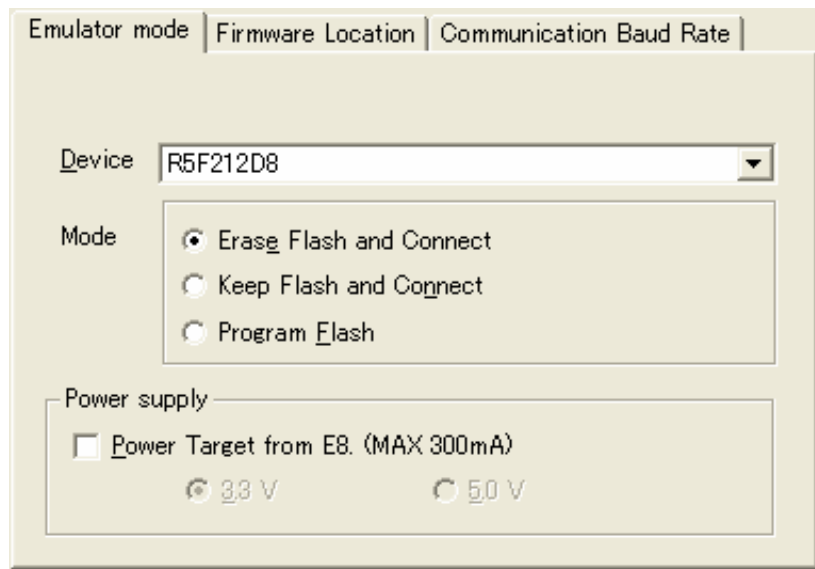


Figure 6.2 [Emulator mode] Tab

### [Device]

Select the name of the MCUs to be used from the [Device] drop-down list box.

### [Mode]

#### - Erase Flash and Connect

When starting up the debugger, erase the flash memory data of the MCUs. The program for the E8 emulator is written at the same time.

#### - Keep Flash and Connect

When starting up the debugger, keep the flash memory data of the MCUs. Note that the area for the E8 emulator program and the E8 emulator used vector area will be changed.

#### - Program Flash

Select this mode when using the E8 emulator as a simple programmer. On downloading write only the user program (The program for the E8 emulator is not written). Therefore, the program cannot be debugged in this mode.

### [Power supply]

When [Power Target from E8. (MAX 300mA)] is checked, power will be supplied to the user system up to 300mA.

## 3. Firmware Location Tab

For details, see “1. Program area for the E8 emulator” of the “Section 5. Notes on Using the E8 Emulator”.

#### 4. Communication Baud Rate Tab

Select communication baud rate between E8 and MCU with the [Communication Baud Rate] tab.  
At ordinary times, select 500000bps (default setting).

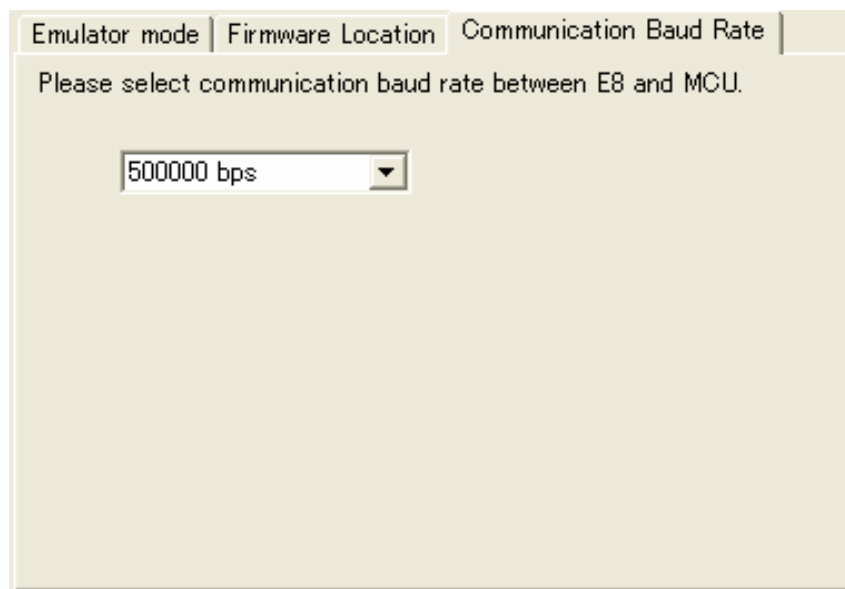


Figure 6.3 [Communication Baud Rate] Tab

## Section 7 Applicable Tool Chain and Partner Tools

With the R8C/Tiny Series E8 emulator, you can debug a module created by the inhouse tool chain and third-party products listed in Table 7.1 below.

Table 7.1 Applicable Tool Chain and Partner Tools

Tool chain	M3T-NC30WA V.5.20 Release 01 or later NC8C V.5.30 Release 01 or later
Partner tools	TASKING M16C C/C++/EC++ Compiler V.2.3r1 or later IAR EWM16C V.2.12 or later



---

E8 Emulator

Additional Document for User's Manual

Notes on Connecting the R8C/2A, R8C/2B, R8C/2C and R8C/2D

Publication Date: February 16, 2007      Rev. 1.00

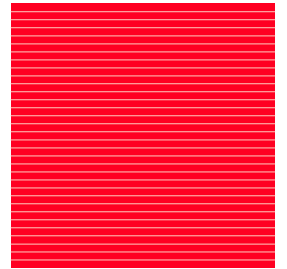
Published by:      Sales Strategic Planning Div.  
Renesas Technology Corp.

Edited by:      Microcomputer Tool Development Department  
Renesas Solutions Corp.

---

© 2007. Renesas Technology Corp. and Renesas Solutions Corp., All rights reserved. Printed in Japan.

# E8 Emulator Additional Document for User's Manual



**Renesas Technology Corp.**  
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan